#6

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

(Attorney Docket No. 13448US02)

In the Patent Application of:

Arthur Abnous et al.

Serial No.: 09/804,082

Filed: March 12, 2001

For: ARCHITECTURE FOR VERY HIGH-SPEED DECISION FEEDBACK SEQUENCE ESTIMATION

Examiner: unassigned

Group Art Unit: 2631

CERTIFICATE OF EXPRESS MAILING

I hereby certify that this correspondence and all correspondence referenced herein are being deposited with the United States Postal Service as Express Mail No. EL649151444US, with sufficient postage, in an envelope addressed to: Asst. Commissioner for Patents, Washington, DC 20231, on December 20, 2001.

Christopher C. Winslade
Reg. No. 36,308

PRELIMINARY AMENDMENT

In view of the Notice of Incomplete Reply (Nonprovisional) mailed October 22, 2001, please amend the above-identified application as follows:

IN THE SPECIFICATION

Please delete the text on page 6 that reads as follows:

"FIG. 2A is a block diagram of one embodiment of the feedforward equalizer constructed in accordance with the present invention."

Please replace the text starting at page 19, line 9 and ending at the very bottom of page 20 (i.e., the text starting with the paragraph on page 19 that reads "FIG 2A is a detailed block diagram..." and continuing through the remainder of page 19 and the entirety of page 20), with the following text (markings indicating changes made are set forth in Appendix A):

-- In one embodiment the FFE 26 includes a precursor filter 28, an inverse partial response filter 30, a noise cancellation stage 32 and a gain stage 34.

The precursor filter, 26 also called a precursor pulse shaping filter, generates a precursor to the input signal 2. This precursor, which is preferably a zero-crossing indicator preceding each sample in the input signal 2, is used for timing recovery by the timing recover module 222 (FIG. 2). The precursor filter 28 is a non-adaptive filter. For ease of implementation and high-speed operation, the precursor filter 28 is preferably a finite impulse response filter having a transfer function of the form $-\gamma + z^{-1}$, with γ equal to 1/16 for short cables (less than 80 meters) and 1/8 for long cables (more than 80 m). The determination of the length of a cable is based on the gain of the coarse PGA 14 of the PGA block 214.

The precursor filter 28 includes a finite impulse response (FIR) filter. In one embodiment of the present invention, the precursor filter 28 also includes a multiplexer and a register. The FIR filter includes a register, a multiplier and an adder. The registers, i.e., the delay elements, are denoted conventionally by z^{-1}. The transfer function of the FIR filter may be expressed as $-\gamma + z^{-1}$ where γ is a programmable constant inputted into the FIR filter via the multiplier. The output y_1 at time sample n of the FIR filter can be expressed in terms of the input sequence x (i.e., the signal 2 outputted from the pair swap multiplexers 224) as $y_1(n) = -\gamma\, x(n) + x(n-1)$.

In this embodiment, the multiplexer provides a value of γ to the FIR filter. This value can be either 1/16 or 1/8, and is selected based on the signal received at the multiplexer select input. This signal is the output of the register. The register has two inputs. The first input is derived from the coarse AGC gain output of the AGC 220 (Figure 2) which is provided to the coarse PGA 14. As implemented in one embodiment, the coarse AGC gain is an unsigned four-bit number. The first input is equal to the most significant bit of the coarse AGC gain. Specifically, the first input is obtained by shifting the coarse AGC gain to the right by three bits and logically AND-ing the shifted word with 1. The second input of the register allows the value of the first input to be loaded into the register. This value is then used by the MUX to select either 1/16 or 1/8 as output. The value 1/16 is selected when the value of the output of the register indicates that the cable connecting the local transceiver to the remote transceiver is short (less than eighty meters). The value 1/8 is selected when the value of the output of the register indicates that the cable connecting the local transceiver to the remote transceiver is long (equal or greater than eighty meters).

The precursor filter 28 preferably includes a register to store the output of the FIR filter and to provide this output to the IPR filter 30 at the next clock pulse. The register prevents any computational delay at the adder of the FIR filter from propagating to the adder of the IPR filter 30. Without this register the concatenation of the two adders may cause a combined computational delay that could exceed a clock period, and this may result in computational errors.--

Please replace the text starting at p. 22, line 19 and ending at page 24, line 7 (i.e., the text starting with the paragraph on page 22 that reads "As shown in FIG. 2A..." and ends immediately <u>before</u> the paragraph on page 24 that reads "The FFE 26..."), with the following text (markings indicating changes made as set forth in Appendix B):

--In one embodiment discussed above, the programmable IPR filter 30 includes an adder, a register and a multiplier. The adder combines the output of the precursor filter 28 with a scaled feedback signal from the output of the IPR filter 30. The scale factor is –K, and is provided by a control signal FFEK. This scale factor is programmable, as previously mentioned. The multiplier multiplies the scale factor with the feedback output of the IPR 30. The transfer function of the IPR 30 is $z^{-1}/(1+Kz^{-1})$. The transfer function would be $1/(1+Kz^{-1})$ if the register is placed on the feedback path instead of the forward path of the filter 30. It is placed on the forward path to prevent any computational delay at the adder from propagating to the downstream adder.

The noise cancellation stage 32 includes an adder and a register. The adder subtracts from the output signal of the IPR filter 30 the noise signals 4, 6, 8, 10, 12 received from the offset canceller 228, NEXT cancellers 230 and echo canceller 232 (FIG. 2). Thus, the output of the adder is a noise-reduced filtered signal. This output is stored in the register and outputted to the gain stage 34 at the next clock pulse.

The gain stage 34 uses a zero-forcing least-mean-squares algorithm to fine-tune the gain of the signal path. The gain stage 34 includes a multiplier and an adaptation circuit. The multiplier scales the output of the noise cancellation stage 32 by the output of the adaptation circuit. Thus, the gain stage 34 adjusts the amplitude of the received signal.

This adjustment provides the adjustment of the gain of the feedforward equalizer 26. The gain stage 34 adjusts the amplitude of the received signal so that it fits in the operational range of the trellis decoder 38 (FIG. 2). This ensures proper operation of the slicer inside the trellis decoder 38 (FIG. 2).

The adaptation circuit includes a multiplier, an adder and a register. The inputs to the multiplier are a 1D component of the tentative decision 44 (FIG. 2) and a 1D component of the slicer error 42 (FIG. 2). The product of these two inputs is shifted to the right by 2 bits. This is the signal $\mu=2^{-2}$. Since the 1D symbols are from the PAM-5 alphabet, the 1D component of the tentative decision 44 can only be –2, -1, 0, 1, 2. The rounded value of slicer error can only be 0 or 1. Thus, the multiplier is actually not a real multiplier.

The adaptation circuit is updated based on a scaled product of the tentative decision and the slicer error. Since the error is also provided to the noise cancellers 228, 230, 232 (FIG. 2), the adaptation circuit is trained on the basis of the error provided to the noise cancellers 228, 230, 232. This allows the adaptation circuit to provide a more accurate gain for the signal path than the PGA 14 (FIG. 2).

The control signal DFEFRZ, when applied, freezes the LMS update of the FFE gain. When it is applied, the register content remains unchanged. The control signal DFERST resets the FFE gain to a value that is decoded from the coarse AGC 220 (FIG. 2) gain. When it is applied, the register content is set to that value.

The output of the gain stage is buffered and delayed by two time periods (two clock pulses) in a register and then outputted.--

Please replace the paragraph starting at page 24, line 16 (i.e., the paragraph on page 24 that starts with the text "Unlike a traditional FFE...)" with the following paragraph (markings indicating changes made are set forth in Appendix C):

--Unlike a traditional FFE which uses adaptive filters for filtering the received signal, the FFE of the present invention uses only non-adaptive filters to filter the signal (it is noted that the adaptation circuit in the gain stage does not filter the received signal). Since the fixed filters are fixed, not adaptive in time, they do not interact with the timing recovery module 222 (FIG. 2). They do not change the phase, hence the pulse shape, of the received signal. Thus, they do not change the sampling phase setting of the timing recovery module 222.--

Please replace the paragraph starting at page 24, line 27 (i.e., the paragraph on page 24 that starts with the text "Another novel feature...") with the following paragraph (markings indicating changes made are set forth in Appendix D):

--Another novel feature of the FFE 26 is that the noise cancellation stage 32 is placed before the adaptive gain stage 34. If the noise cancellation stage is placed after the gain stage, then the impulse responses of the cancellers 228, 230, 232 will be affected by the gain of the gain stage for the following reason. The coefficients of the cancellers are trained for certain gain value. When the gain changes, the coefficients of the cancellers are no longer correct and need to be retrained. Thus, because of this interaction between the gain stage and the cancellers, the startup will be unreliable. Therefore, the placement of the noise cancellation stage 32 before the gain stage 34 causes the feedback loop between the adaptive gain stage 34 and the cancellers 228, 230, 232 to be de-coupled. This in turn allows the startup to be robust. When the echo, NEXT, and offset cancellation is done before the gain stage, as discussed above, the coefficients of the echo, NEXT and offset cancellers do not need to change in response to gain changes, as discussed previously. However, it is important to note that, unless special compensation logic is added, the gain of the LMS update algorithm for the cancellers would change. This in turn would cause the speed of convergence of the cancellers to change when the gain of the FFE changes. In some cases (when the gain of the FFE is large) it would even cause instabilities in the adaptation algorithm for the cancellers. To prevent this from happening, the cancellers are adapted using the "normalized adaptation error" 42enc (FIG. 15) instead of the slicer error 42ph (FIG. 15) or the adaptation error 42dfe (FIG. 15). An exact

normalization would require that the normalized adaptation error 42enc be computed by dividing the adaptation error 42dfe by the gain of the gain stage 34. However a true divider circuit is complex and difficult to implement at high speed. Therefore, an approximate division is used to compute the normalized adaptation error 42enc. The approximate division is done using only the 4 most significant bits (MSBs) of the gain of the gain stage 34 (the gain is treated as a U13.8 quantity, i.e., an unsigned number having 13 bits with 8 bits after the decimal point). This approximate division is as follows:

if the MSB = 1	Normalized Adaptation Error = Adaptation Error shifted to the right by 1 bit;
else if the 2nd MSB == 1	Normalized Adaptation Error = Adaptation Error
else if the 3rd MSB == 1	Normalized Adaptation Error = Adaptation Error shifted to the left by 1 bit;
else	Normalized Adaptation Error = Adaptation Error shifted to the left by 2 bits.--

REMARKS

Applicant has deleted reference in the specification to omitted FIG. 2A.

Should the Examiner have any questions regarding this submission, Applicant respectfully requests the Examiner telephone the undersigned at 312-775-8000.

A Notice of Allowability is courteously solicited.

Respectfully,

Christopher C. Winslade
Registration No. 36,308

Date: December 20, 2001

McAndrews, Held & Malloy
500 W. Madison St.,
34th Floor
Chicago, IL 60661
(312) 775-8000

APPENDIX A

[FIG. 2A is a detailed block diagram of an exemplary embodiment of the FFE 38. This] In one embodiment [of] the FFE [38] 26 includes a precursor filter 28, an inverse partial response filter 30, a noise cancellation stage 32 and a gain stage 34.

The precursor filter [28] 26, also called a precursor pulse shaping filter, generates a precursor to the input signal 2. This precursor, which is preferably a zero-crossing indicator preceding each sample in the input signal 2, is used for timing recovery by the timing recover module 222 (FIG. 2). The precursor filter 28 is a non-adaptive filter. For ease of implementation and high-speed operation, the precursor filter 28 is preferably a finite impulse response filter having a transfer function of the form $-\gamma + z^{-1}$, with γ equal to 1/16 for short cables (less than 80 meters) and 1/8 for long cables (more than 80 m). The determination of the length of a cable is based on the gain of the coarse PGA 14 of the PGA block 214.

The precursor filter 28 includes a finite impulse response (FIR) filter [122]. In one embodiment of the present invention, the precursor filter 28 also includes a multiplexer [132] and a register [136]. The FIR filter [122] includes a register [124], a multiplier [126] and an adder [128]. The registers, i.e., the delay elements, are denoted conventionally by z^{-1}. The transfer function of the FIR filter [122, as shown in Figure 2A,] may be expressed as $-\gamma + z^{-1}$ where γ is a programmable constant inputted into the FIR filter [122] via the multiplier [126]. The output y_1 at time sample n of the FIR filter [122] can be expressed in terms of the input sequence x (i.e., the signal 2 outputted from the pair swap multiplexers 224) as $y_1(n) = -\gamma\, x(n) + x(n-1)$.

In [the] this embodiment [shown in Figure 2A], the multiplexer [132] provides a value of γ to the FIR filter [122]. This value can be either 1/16 or 1/8, and is selected based on the signal received at the multiplexer select input. This signal is the output [134] of the register [136]. The register [136] has two inputs [138 and 140]. The first input [138] is derived from the coarse AGC gain output of the AGC 220 (Figure 2) which is provided to the coarse PGA 14. As implemented in one embodiment, the coarse AGC gain is an unsigned four-bit number. The first input [138] is equal to the most significant bit of the coarse AGC gain. Specifically, the first input [138] is obtained by shifting the coarse AGC gain to the right by three bits and logically AND-ing the shifted word with 1. The second

input [140] of the register [136] allows the value of the first input [138] to be loaded into the register [136]. This value is then used by the MUX [132] to select either 1/16 or 1/8 as output. The value 1/16 is selected when the value of the output of the register [136] indicates that the cable connecting the local transceiver to the remote transceiver is short (less than eighty meters). The value 1/8 is selected when the value of the output of the register [136] indicates that the cable connecting the local transceiver to the remote transceiver is long (equal or greater than eighty meters).

The precursor filter 28 preferably includes a register [130] to store the output of the FIR filter [122] and to provide this output to the IPR filter 30 at the next clock pulse. The register [130] prevents any computational delay at the adder [128] of the FIR filter [122] from propagating to the adder [142] of the IPR filter 30. Without this register [130] the concatenation of the two adders [128, 142] may cause a combined computational delay that could exceed a clock period, and this may result in computational errors.

APPENDIX B

[As shown in FIG. 2A,] In one embodiment discussed above, the programmable IPR filter 30 includes an adder [142], a register [144] and a multiplier [146]. The adder [142] combines the output of the precursor filter 28 with a scaled feedback signal from the output of the IPR filter 30. The scale factor is –K, and is provided by a control signal FFEK. This scale factor is programmable, as previously mentioned. The multiplier [146] multiplies the scale factor with the feedback output of the IPR 30. The transfer function of the IPR 30[, as shown,] is $z^{-1}/(1+Kz^{-1})$. The transfer function would be $1/(1+Kz^{-1})$ if the register [144] is placed on the feedback path instead of the forward path of the filter 30. It is placed on the forward path to prevent any computational delay at the adder [142] from propagating to the downstream adder [148].

The noise cancellation stage 32 includes an adder [148] and a register [150]. The adder [148] subtracts from the output signal [145] of the IPR filter 30 the noise signals 4, 6, 8, 10, 12 received from the offset canceller 228, NEXT cancellers 230 and echo canceller 232 (FIG. 2). Thus, the output [149] of the adder [148] is a noise-reduction filtered signal. This output [149] is stored in the register [150] and outputted to the gain stage 34 at the next clock pulse.

The gain stage 34 uses a zero-forcing least-mean-squares algorithm to fine-tune the gain of the signal path. The gain stage 34 includes a multiplier [152] and an adaptation circuit [154]. The multiplier [152] scales the output [151] of the noise cancellation stage 32 by the output [161] of the adaptation circuit [154]. Thus, the gain stage 34 adjusts the amplitude of the received signal [151]. This adjustment provides the adjustment of the gain of the feedforward equalizer 26. The gain stage 34 adjusts the amplitude of the received signal [151] so that it fits in the operational range of the trellis decoder 38 (FIG. 2). This ensures proper operation of the slicer inside the trellis decoder 38 (FIG. 2).

The adaptation circuit [154] includes a multiplier [156], an adder [158] and a register [160]. The inputs to the multiplier [156 is] are a 1D component [44A] of the tentative decision 44 (FIG. 2) and a 1D component of the slicer error 42 (FIG. 2). The product of these two inputs is shifted to the right by 2 bits. This is [indicated in FIG. 2A by] the signal $\mu=2^{-2}$. Since the 1D symbols are from the PAM-5 alphabet, the 1D component [44A] of the

tentative decision 44 can only be –2, -1, 0, 1, 2. The rounded value of slicer error [42A] can only be 0 or 1. Thus, the multiplier [156] is actually not a real multiplier.

The adaptation circuit [154] is updated based on a scaled product of the tentative decision [44A] and the slicer error [42A]. Since the error [42A] is also provided to the noise cancellers 228, 230, 232 (FIG. 2), the adaptation circuit [154] is trained on the basis of the error provided to the noise cancellers 228, 230, 232. This allows the adaptation circuit [154] to provide a more accurate gain for the signal path than the PGA 14 (FIG. 2).

The control signal DFEFRZ, when applied, freezes the LMS update of the FFE gain. When it is applied, the register [160] content remains unchanged. The control signal DFERST resets the FFE gain to a value that is decoded from the coarse AGC 220 (FIG. 2) gain. When it is applied, the register [160] content is set to that value.

The output [153] of the gain stage is buffered and delay by two time periods (two clock pulses) in a register [162] and then outputted.

APPENDIX C

Unlike a traditional FFE which uses adaptive [filter] filters for filtering the received signal, the FFE of the present invention uses only non-adaptive filters to filter the signal (it is noted that the adaptation circuit [154] in the gain stage does not filter the received signal). Since the fixed filters [20 and 30] are fixed, not adaptive in time, they do not interact with the timing recovery module 222 (FIG. 2). They do not change the phase, hence the pulse shape, of the received signal. Thus, they do not change the sampling phase setting of the timing recovery module 222.

APPENDIX D

Another novel feature of the FFE 26 is that the noise cancellation stage 32 is placed before the adaptive gain stage 34. If the noise cancellation stage is placed after the gain stage, then the impulse responses of the cancellers 228, 230, 232 will be affected by the gain of the gain stage for the following reason. The coefficients of the cancellers are trained for certain gain value. When the gain changes, the coefficients of the cancellers are no longer correct and need to be retrained. Thus, because of this interaction between the gain stage and the cancellers, the startup will be unreliable. Therefore, the placement of the noise cancellation stage 32 before the gain stage 34 causes the feedback loop between the adaptive gain stage 34 and the cancellers 228, 230, 232 to be de-coupled. This in turn allows the startup to be robust. When the echo, NEXT, and offset cancellation is done before the gain stage, as discussed above [in FIG. 2A], the coefficients of the echo, NEXT and offset cancellers do not need to change in response to gain changes, as discussed previously. However, it is important to note that, unless special compensation logic is added, the gain of the LMS update algorithm for the cancellers would change. This in turn would cause the speed of convergence of the cancellers to change when the gain of the FFE changes. In some cases (when the gain of the FFE is large) it would even cause instabilities in the adaptation algorithm for the cancellers. To prevent this from happening, the cancellers are adapted using the "normalized adaptation error" 42enc (FIG. 15) instead of the slicer error 42ph (FIG. 15) or the adaptation error 42dfe (FIG. 15). An exact normalization would require that the normalized adaptation error 42enc be computed by dividing the adaptation error 42dfe by the gain [161] of the gain stage 34. However a true divider circuit is complex and difficult to implement at high speed. Therefore, an approximate division is used to compute the normalized adaptation error 42enc. The approximate division is done using only the 4 most significant bits (MSBs) of the gain of

the gain stage 34 [161] (the gain [161] is treated as a U13.8 quantity, i.e., an unsigned number having 13 bits with 8 bits after the decimal point). This approximate division is as follows:

if the MSB = 1	Normalized Adaptation Error = Adaptation Error shifted to the right by 1 bit;
else if the 2nd MSB == 1	Normalized Adaptation Error = Adaptation Error
else if the 3rd MSB == 1	Normalized Adaptation Error = Adaptation Error shifted to the left by 1 bit;
else	Normalized Adaptation Error = Adaptation Error shifted to the left by 2 bits.